UNITED STATES PATENT APPLICATION

FOR

CONTROL OF AUTHENTICATION DATA RESIDING IN A NETWORK DEVICE

INVENTORS:

Norman C. Chou
Olivier Cremel

Prepared by:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CALIFORNIA 90025
(408) 720-8598

Attorney's Docket No. 05288.P017

"Express Mail" mailing label number: EL431888057US
Date of Deposit: January 24, 2002
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Michelle Begay
(Typed or printed name of person mailing paper or fee)

Michelle Begay
(Signature of person mailing paper or fee)

January 24, 2002
(Date signed)

CONTROL OF AUTHENTICATION DATA RESIDING IN A NETWORK DEVICE

FIELD OF THE INVENTION

[0001] The present invention relates generally to the field of data communications and, more specifically, to controlling authentication data residing in a interconnect device.

BACKGROUND OF THE INVENTION

[0002] Existing networking and interconnect technologies have failed to keep pace with the development of computer systems, resulting in increased burdens being imposed upon data servers, application processing and enterprise computing. This problem has been exasperated by the popular success of the Internet. A number of computing technologies implemented to meet computing demands (e.g., clustering, fail-safe and 24X7 availability) require increased capacity to move data between processing nodes (e.g., servers), as well as within a processing node between, for example, a Central Processing Unit (CPU) and Input/Output (I/O) devices.

[0003] With a view to meeting the above described challenges, a new interconnect technology, called the InfiniBand™, has been proposed for interconnecting processing nodes and I/O nodes to form a System Area Network (SAN). This architecture has been designed to be independent of a host Operating System (OS) and processor platform. The InfiniBand™ Architecture (IBA) is

centered around a point-to-point, switched IP fabric whereby end node devices (e.g., inexpensive I/O devices such as a single chip SCSI or Ethernet adapter, or a complex computer system) may be interconnected utilizing a cascade of switch devices. The InfiniBand™ Architecture is defined in the InfiniBand™ Architecture Specification Volume 1, Release 1.0, released October 24, 2000 by the InfiniBand Trade Association. The IBA supports a range of applications ranging from back plane interconnect of a single host, to complex system area networks, as illustrated in **Figure 1** (prior art). In a single host environment, each IBA switched fabric may serve as a private I/O interconnect for the host providing connectivity between a CPU and a number of I/O modules. When deployed to support a complex system area network, multiple IBA switch fabrics may be utilized to interconnect numerous hosts and various I/O units.

[0004] Within a switch fabric supporting a System Area Network, such as that shown in **Figure 1**, there may be a number of devices having multiple input and output ports through which data (e.g., packets) is directed from a source device to a destination device. Such devices include, for example, switches, routers, repeaters and adapters (exemplary interconnect devices). In addition to multiple communication ports directing external data packets, an interconnect device such as a switch typically includes a management port. Each sub-network (subnet) is managed by at least one Subnet Manager. A Subnet Manager resides either on an endnode or on an interconnect device and can be implemented either in hardware or

software. The Subnet Manager performs its managing functions by communicating with the management port using InfiniBand™ Subnet Management Packets.

[0005] Subnet Management Packets (SMPs) are used to initialize and configure switches and other interconnect devices, and are therefore considered to participate in privileged operations. As a result, a mechanism is provided to authorize subnet management operations by comparing authentication data included in a SMP with authentication data stored in a destination port. The authentication data includes a Management Key (e.g., the InfiniBand™ Management Key). The Management Key is associated with several attributes that may affect the authorization of subnet management operations. For example, these attributes may include a protection attribute that identifies levels of protection required for specific subnet management operations and an expiration attribute that allows the management key to "expire" if the management key is lost. The expiration of the management key attribute is not permitted if the expiration attribute is set to zero. Accordingly, a problem may arise when the management key is lost or becomes contaminated while the expiration attribute is equal to zero.

SUMMARY OF THE INVENTION

[0006] Methods and systems for supporting management operations associated with an interconnect device are described. According to one aspect of the present invention, an exemplary system includes a port of the interconnect device to maintain authentication data that facilitates authorization of a management operation and a configuration switch coupled to the port to generate a reset signal in response to an operator's command. The port is operable to reset the authentication data upon receiving the reset signal from the configuration switch.

[0007] According to another aspect of the present invention, an exemplary method includes detecting that a reset of authentication data maintained by a management port of an interconnect device is necessary, informing the operator that the reset is required and refraining from sending subnet management packets (SMPs) to the management port until receiving a message from the operator indicating that the authentication data maintained by the management port has been reset. The method further includes sending to the management port an update SMP that includes a request to set authentication data residing in each unit of the interconnect device to a specific update value.

[0008] Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0010] Figure 1 is a diagrammatic representation of a System Area Network, according to the prior art, as supported by a switch fabric;

[0011] Figure 2 is a block diagram of a system for supporting management operations associated with an interconnect device, according to one embodiment of the present invention;

[0012] Figure 3 is a block diagram of a system for maintaining authentication data in a management port of an interconnect device, according to one embodiment of the present invention;

[0013] Figure 4 is an exemplary datagram of a Subnet Management Packet (SMP) received by a decoder of **Figure 3,** according to one embodiment of the present invention;

[0014] Figure 5 is a flow diagram of a method for controlling authentication data residing in an interconnect device, according to one embodiment of the present invention;

[0015] Figure 6 is a flow diagram of a method for supporting management operations associated with an interconnect device, according to one embodiment of the present invention; and

[0016] Figures 7A and 7B illustrate the operation of some embodiments of the present invention using two exemplary scenarios.

DETAILED DESCRIPTION

[0017] Methods and systems to support management operations associated with an interconnect device are described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details.

[0018] Note also that embodiments of the present description may be implemented not only within a physical circuit (e.g., on semiconductor chip) but also within machine-readable media. For example, the circuits and designs discussed above may be stored upon and/or embedded within machine-readable media associated with a design tool used for designing semiconductor devices. Examples include a netlist formatted in the VHSIC Hardware Description Language (VHDL) language, Verilog language or SPICE language. Some netlist examples include: a behavioral level netlist, a register transfer level (RTL) netlist, a gate level netlist and a transistor level netlist. Machine-readable media also include media having layout information such as a GDS-II file. Furthermore, netlist files or other machine-readable media for semiconductor chip design may be used in a simulation environment to perform the methods of the teachings described above.

[0019] Thus, it is also to be understood that embodiments of this invention may be used as or to support a software program executed upon some form of processing core (such as the CPU of a computer) or otherwise implemented or realized upon or within a machine-readable medium. A machine-readable medium

includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0020] For the purposes of the present invention, the term "interconnect device" shall be taken to include switches, routers, repeaters, adapters, or any other device that provides interconnect functionality between nodes. Such interconnect functionality may be, for example, module-to-module or chassis-to-chassis interconnect functionality. While an exemplary embodiment of the present invention is described below as being implemented within a switch deployed within an InfiniBand architectured system, the teachings of the present invention may be applied to any interconnect device within any interconnect architecture.

[0021] Figure 2 is a block diagram of a system 200 for supporting management operations associated with an interconnect device, according to one embodiment of the present invention. System 200 includes a sub-network (subnet) 210 and a subnet manager 206 coupled to subnet 210 that configures and manages subnet 210. Subnet 210 includes one or more interconnect devices 202 such as switches, routers, repeaters, adapters, etc. Subnet manager 206 communicates with each interconnect device 202 to perform managing functions which include, for example, initialization and configuration of interconnect device 202. The

communication between subnet manager 206 and interconnect device 202 is accomplished using data packets designated to perform managing operations. In one embodiment, these data packets are InfiniBand™ Subnet Management Packets as will be described in more detail below.

[0022] Because the data packets sent by subnet manager 206 participate in privileged operations, an authorization mechanism is provided to prevent unauthorized entities from performing such operations. The authorization mechanism may be supported by subnet manager 206 which places authentication data in interconnect device 202 during its configuration and also stores the authentication data in a subnet manager database 208. Subsequently, when creating a data packet with a request to perform a managing operation, subnet manager 206 retrieves the authentication data associated with interconnect device 202 from database 208 and includes this authentication data into the data packet. Once interconnect device 202 receives the data packet from subnet manager 206, interconnect device 202 compares the authentication data included in the data packet with the authentication data stored locally in interconnect device 202 to determine whether the data packet was sent by an authorized entity. If both data items match (or the authentication data residing in the interconnect device 202 is set to a value that does not require an authentication check), interconnect device 202 performs the operation requested by the data packet and sends a response back to subnet manager 206. Alternatively, if a mismatch of the data items occurs, interconnect device 202 discards the data packet without sending any response back.

[0023] In some situations, the authentication data associated with interconnect device 202 may be lost. For example, a failure of subnet manager 206 may occur before subnet manager 206 shares the authentication data with a successor (e.g., a backup subnet manager). In this situation, a conventional approach for recovering the authentication data is based on an expiration attribute associated with the authentication data. Specifically, when the value of the expiration attribute permits, the authentication data residing in interconnect device 202 can expire, thereby returning interconnect device 202 into a state that allows the successor of interconnect device 202 to establish new authentication data. However, some values of the expiration attribute provide for the indefinite duration of the authentication data, i.e., they do not allow the authentication data to expire. As a result, if a violation of the authentication data occurs at the time when its expiration is not permitted, subnet manager 206 or its successor can no longer manage interconnect device 202.

[0024] The present invention provides a mechanism that allows subnet manager 206 or any other authorized entity to regain control over interconnect device 202 at any time. Specifically, a configuration switch 204 is provided which is responsible for receiving an operator's command to reset the authentication data stored in interconnect device 202 and generating a rest signal in response to the operator's command. Interconnect device 202 receives the reset signal from configuration switch 204 and resets the authentication data, thereby returning interconnect device 202 into a state that allows subnet manager 206 or any other

authorized entity to establish new authentication data within interconnect device 202. In one embodiment, the configuration switch 204 includes a pin located outside of interconnect device 202. A change of the pin's state (i.e., the pin is being either set or cleared) causes a generation of a reset signal.

[0025] In one embodiment, configuration switch 204 is coupled to each interconnect device within subnet 210. Alternatively, a separate configuration switch is provided for each interconnect device within subnet 210.

In one embodiment, configuration switch 204 is coupled to a management port of interconnect device 202 that stores multiple copies of authenticated data. **Figure 3** is a block diagram of a system 300 for maintaining authentication data in a management port of an interconnect device, according to one embodiment of the present invention.

[0026] Referring to **Figure 3**, management port 302 includes a set of agents that perform various functions. One of the agents is an initialization agent 310 that is responsible for controlling the initialization of the interconnect device. Initialization agent 310 stores authentication data and a set of associated attributes. In one embodiment, the authentication data consists of a management key (e.g., an InfiniBand™ M_Key), and the associated attributes include a management key protection attribute (e.g., InfiniBand™ M_KeyProtectBits) and an expiration attribute (e.g., InfiniBand™ M_KeyLeasePeriod). The protection attribute defines levels of protection for operations requested by subnet manager 332. The expiration attribute allows the management key to expire when the management key is lost or

contaminated. Initialization agent 310 knows which units within the interconnect device store authentication data and communicates with these units when the update of the authentication data is requested.

[0027] Another agent residing in management port 302 is a processor subsystem interface 312. The processor subsystem interface 312 is coupled to a processor subsystem 320 via a processor bus 318 and/or an independent non-volatile random access memory (NVRAM) 326 via a bus 328 (e.g., an Inter-IC (I2C) bus). Processor subsystem 320 includes a processor 322 and may or may not include a non-volatile storage device 324. Configuration information associated with the interconnect device may be stored either in NVRAM 326 or storage device 324. The configuration information includes a copy of the authentication data (e.g., the management key) and associated attributes (including the protection attribute and the expiration attribute). Processor subsystem interface 312 knows in which device the configuration information resides and requests the configuration information or its portions (e.g., the management key) from the appropriate storage device when needed.

[0028] Yet other agents residing in management port 302 include a decoder 304 and a subnet management agent (SMA) 308. Decoder 304 is responsible for decoding and dispatching data packets received at management port 302 to destination agents within management port 302. SMA 308 is a targeting destination agent for subnet management packets (SMPs) sent by a subnet manager 332. Each of decoder 304 and SMA 308 includes a copy of the management key and associated

attributes. When decoder 304 determines that a data packet being decoded is a SMP, decoder 304 compares its copy of the management key with the management key included in the packet. If the two management keys match, decoder 304 forwards the SMP to SMA 308. Otherwise, decoder 304 discards the SMP. This authentication check is not performed when the decoder's copy of the management key is set to zero.

[0029] In one embodiment, a configuration switch 306 is coupled to decoder 304 to allow a reset of the decoder's copy of the management key when needed. For example, the mismatch between the decoder's management key and the management key included in a SMP may be unintended and caused by malfunctioning of the interconnect device and/or subnet manager 332. Then, an operator can issue a command to reset the management key (e.g., by pushing a button associated with switch 306). In one embodiment, subnet manager 332 informs the operator (e.g., via any visual or audible means) that the reset of the management key is needed. In another embodiment, the interconnect device itself may be operable to inform the operator that the reset is required. In yet another embodiment, the operator can issue the reset command via configuration switch 306 without any notification from the interconnect device or subnet manager 332 (e.g., the reset may be a part of installation of the interconnect device).

[0030] Configuration switch 306 responds to the operator's command by generating a reset signal that causes decoder 304 to "zap" (i.e., to set to zero) its copy of the management key. While the decoder's copy of the management key is equal

to zero, no authentication check will be performed for SMPs received at management port 302, thus allowing the communication between subnet manager 332 and SMA 308 to resume.

[0031] In one embodiment, once decoder 304 receives the reset signal, it resets its copy of the management key and forwards the reset signal to initialization agent 310, which in turn resets its management key and sends a request to reset a copy of the management key to each component of the interconnect device that stores such a copy. In another embodiment, decoder 304 only resets its copy of the management key and does not communicate the reset signal to other agents of management port 302.

[0032] In one embodiment, subnet manager 332 receives a resume command from the operator once the decoder's management key is reset. The resume command indicates that the communication between subnet manager 332 and management port 302 can be resumed. In response to the resume command, subnet manager 332 sends a new SMP to management port 302 with a request to update the management key residing in management port 302 to the value of the management key that is stored in the subnet manager's database. This process will be described in greater detail below.

[0033] **Figure 4** is an exemplary datagram of a SMP received by decoder 302 of **Figure 3,** according to one embodiment of the present invention.

[0034] Referring to **Figure 4**, the positions of each field within the packets is provided in bits words. When there are two numbers, the number in parenthesis is

given for a packet without a global router header (GRH), and the other number is given for a packet that does not include a GRH.

[0035] Decoder 302 determines whether the packet is a SMP using a virtual lane (VL) identifier 404 and a destination queue pair (DQP) identifier 406. VL identifier 404 specifies whether this packet is a VL 15 packet or a non-VL 15 packet. VLs are, in one embodiment, independent data streams that are supported by a common physical link. A VL may represent a set of transmit and receive buffers in a port. VL15 is reserved exclusively for subnet management packets (SMPs). DQP identifier 406 identifies a target destination queue pair. A queue pair is used to queue up a set of instructions that the hardware executes. A queue pair consists of a queue for send operations and a queue for receive operations. VL 15 packets must use queue pair 0 and non-VL 15 packets can use any other queue pairs except queue pair 0. Further details regarding the concepts of "virtual lanes" and "queue pairs" are provided in the InfiniBand™ Architecture Specification, Volume 1, October 24, 2000.

[0036] Decoder 304 decides that the packet is a SMP if VL identifier 404 is equal to fifteen and DQP identifier 406 is equal to zero. A management class 408 identifies a particular agent that should process the packet. For a SMP, management class 408 stores the value associated with a SMA. A method 410 identifies an operation (e.g., read or write) requested by the packet. An attribute identifier 412 and an attribute modifier 414 are used to identify the location of the requested operation. M_Key fields 416 store a 64-bit management key that needs to be

compared with the decoder's copy of the management key to authenticate the SMP at the management port.

[0037] It should be noted that various other fields in the packet can be used to extract the information required by system 300. In addition, incoming packets may have a variety of other formats and fields that decoder 304 may use to extract the required information.

[0038] **Figure 5** is a flow diagram of a method 500 for controlling authentication data residing in an interconnect device, according to one embodiment of the present invention. Method 500 is performed by processing logic, which may comprise hardware, software, or a combination of both. The processing logic of method 500 is implemented in a management port of the interconnect device (e.g., in decoder 304 of **Figure 3**).

[0039] Method 500 begins with the processing logic receiving a reset signal from a configuration switch coupled to the management port (processing block 504). The reset signal indicates that an operator has requested to reset authentication data stored in a decoder of the management port. As described in more detail above, the authentication data facilitates the authorization of management operations associated with the interconnect device. In one embodiment, the authentication data is represented as a management key.

[0040] At processing block 506, the processing logic resets the decoder's copy of the authentication data. Because no authentication of an incoming data packet is performed when the authentication data is set to zero, the reset of the decoder's copy

of the authentication data allows a subnet manager to communicate with the management port when their authentication data does not match.

[0041] Next, at processing block 508, the processing logic receives a data packet from the subnet manager that includes a request to update the authentication data residing in the interconnect device with a new value that matches the value of the management key maintained by the subnet manager. In one embodiment, prior to receiving the data packet with the request to update the authentication data, the processing logic receives the subnet manager's data packet with a request to read the authentication data residing in the management port. The processing logic decodes this data packet and sends it to a subnet management agent (SMA) residing in the management port. The SMA reads the authentication data maintained by the initialization agent, generates a response including this authentication data and sends the response to the subnet manager. The subnet manager may then use this authentication data as an update value in the subsequent data packet that includes the update request.

[0042] Afterwards, at processing block 510, the processing logic sets the decoder's copy of the management key to the new value. In one embodiment, the update of the decoder's copy of the management key is performed upon receiving an update request from an initialization agent of the management key. Specifically, once the decoder receives a valid data packet from the subnet manager, it decodes the data packet and sends the data packet to a subnet management agent (SMA) residing in the management port. The SMA determines that the data packet includes

a request to update the authentication data, updates its copy of the authentication data, and notifies the initialization agent about this request. The initialization agent updates its own authentication data and sends a command to update a corresponding copy of the authentication data to each unit of the interconnect device that stores such a copy, including the decoder. As a result, the mismatch between the authentication data maintained by the subnet manager and the authentication data maintained by the interconnect device is corrected, and normal authentication operations can resume.

[0043] Figure 6 is a flow diagram of a method 600 for supporting management operations associated with an interconnect device, according to one embodiment of the present invention. Method 600 is performed by processing logic, which may comprise hardware, software, or a combination of both. The processing logic of method 500 is implemented in a subnet manager such as a subnet manager 206 of **Figure 2.**

[0044] Method 600 begins with the processing logic detecting that a reset of authentication data (e.g., a management key) maintained by a management port of the interconnect device is required (processing block 604). In one embodiment, the reset requirement is detected upon receiving a trap indicating that the management port has invalidated an initial data packet (e.g., a SMP) sent to the management port by the subnet manager due to a violation of the authentication data. In one embodiment, the trap is issued when the management port detects a mismatch between the authentication data stored in the management port and the

authentication data included in the initial data packet, and an expiration attribute associated with the authentication data is set to the value (e.g., a zero) that prevents the expiration of the authentication data as explained in greater detail above.

[0045] In another embodiment, the reset requirement is detected in response to the following sequence of events:

(a) the initial data packet is sent to the management port;

(b) no response to the initial packet is received from the management port for a predefined time period; and

(c) the initial data packet is re-sent a predetermined number of times without receiving any response from the management port.

Based on this sequence of events, the processing logic decides that the failure of the management port to respond is probably caused by the violation of the authentication data.

[0046] It should be noted that any other mechanism known in the art can be used to allow the subnet manager to detect the violation of the authentication data.

[0047] Next, at processing block 606, the processing logic notifies the operator that the reset is required. The notification may be in any visual form (e.g., a message on a display device, a red light of an LED, etc.) or audio form (e.g., a particular sound, a sequence of sounds, etc.).

[0048] Once the processing logic determines that the reset is required, it prevents the transmission of data packets to the management port (processing logic 608) until receiving a message from the operator that indicates that the

authentication data maintained by the management port has been reset (processing block 610). Subsequent to the operators' message, the processing logic sends to the management port an update data packet with a request to set the authentication data maintained by the interconnect device to an update value (processing block 612). In one embodiment, the update value is the value stored in a database of the subnet manager. In another embodiment, prior to sending the update data packet, the processing logic determines the update value. In this embodiment, the operator's reset command causes only the reset of the decoder's copy of the authentication data; other agents of the management port still store valid authentication data, as described in greater detail above. In this embodiment, once the processing logic receives the operator's message indicating that the reset has been performed, the processing logic sends to the management port a read data packet requesting the current value of the authentication data maintained by the management port. In one embodiment, this data packet is processed by the SMA of the management port that queries the initialization agent for the current value of the authentication data and sends a response with this value back to the subnet manager. The processing logic then updates the database of the subnet manager with the received value of the authentication data and also uses this value as an update value in the update data packet that is sent to the management port at processing block 612.

[0049] Figures 7A and 7B illustrate the operation of some embodiments of the present invention using two exemplary scenarios.

[0050] Referring to **Figure 7A**, a master subnet manager 706 manages subnet 710 that includes an interconnect device 702. Master subnet manager 706 performs its managing functions by communicating with a SMA that resides in a management port of interconnect device 702 using SMPs. As described above, each SMP includes a management key that is compared with a management key stored in a decoder of the management port unless the decoder's management key is set to zero.

[0051] The management key is stored in the management port with a set of associated attributes including an expiration attribute. The expiration attributes facilitates the expiration of the management key. For example, according to the InfiniBand™Architecture, if the management key included in the SMP does not match the management key stored in the decoder, the expiration attribute (referred to as "M_KeyLeasePeriod") gets set to a certain time period at the end of which the management key will expire. Accordingly, although the current SMP that includes the mismatched management key is discarded, an SMP received at the management port at a later time (after the management key has expired) will not undergo the authentication check and will be processed by the SMA as a valid SMP. This approach allows the recovery of the management key when the management key is lost or contaminated. However, this approach is used only if the expiration attribute is not equal to zero because the expiration of the management key is not permitted when the expiration attribute is set to zero.

[0052] Master subnet manager 706 maintains a database 710 where the management key associated with interconnect device 702 is stored. When master

subnet manager 706 inadvertently goes away, a transition to a backup subnet manager 708 takes place. During this transition, the correct management key may be lost (e.g., database 710 may be lost or its data may get contaminated), leaving backup subnet manager 708 with an incorrect management key. Subsequently, when backup subnet manager 708 sends an SMP to the SMA of the management port, the authentication check performed by the decoder will fail. If at this time, the expiration attribute is set to zero at this time, backup subnet manager 708 will detect that the management key is violated using one of the mechanisms described above and will notify the operator that the reset of the management key residing in the management port is needed. The operator will then send a reset command to the management port via a configuration switch 204.

[0053] In addition to the reset command, the operator will submit a message to backup subnet manager 708 (e.g., by pushing a corresponding button) indicating that the reset of the management key has been completed. In response, backup subnet manager 708 will recover the management key using one of the approaches described above.

[0054] Referring to **Figure 7B**, the size of subnet 730 is increased by adding an interconnect device 722 initially included in subnet 720. The management key associated with interconnect device 722 is known to a subnet manager 724 but not a subnet manager 736. In one embodiment, to avoid the mismatch of management keys, the installation procedure requires the operator performing the addition of interconnect device 726 to subnet 730 to reset the management key 726 using a

configuration switch 728. In another embodiment, the installation procedure does not require the reset of the management key. Instead, subnet manager 736 detects that such reset is required after the installation of interconnect device 726 is completed and notifies the operator about the reset requirement as described in greater detail above.

[0055] In one embodiment, a reset signal generated by configuration switch 728 will zap each management key stored in interconnect device 726. Alternatively, a reset signal generated by configuration switch 728 will only zap the management key of the decoder. In either embodiment, once subnet manager 736 receives a message indicating that the reset of the management key has been performed, subnet manager 736 sends an update SMP to set the management keys of interconnect device 726 to a desired value.

[0056] The scenarios illustrated in **Figures 7A and 7B** are based on the management key violations that were caused by external events (i.e., events that did not take place within the interconnect device). However, a management key violation may also occur due to an internal cause. For example, referring back to **Figure 3**, in one embodiment, a processor bus 318 is 16-bit wide and an I2C bus 328 is 1-bit wide while the management key is 64-bit long. When subnet manager 332 requests an update of each management key within the interconnect device, the update of the management key stored either in NVRAM 326 or storage device 324 involves several operations (because only a portion of the update management key can be passed during one operation via a corresponding bus). If these operations are

interrupted before they are completed (e.g., due to a reset or a malfunction), NVRAM 326 or storage device 324 will store an incorrect management key that may be passed to the other units of the interconnect device during a reboot, causing a mismatch between the management key maintained by the interconnect device and the management key maintained by subnet manager 332.

[0057] In another example, a contamination of the management key may happen when NVRAM 326 is switched to storage device 324 (or vice versa) due to a problem with NVRAM 326. If the management key stored in NVRAM 326 has been recently updated, storage device 324 may not store this current version of the management key but rather store a previous, old version of the management key. During a reboot, this old version of the management key may be passed to the other units of the interconnect device, causing a mismatch between the management key maintained by the interconnect device and the management key maintained by subnet manager 332.

[0058] In each of these two examples, if at the time the mismatch is detected, the expiration attribute is set to zero, the decoder's copy of the management key will need to be reset via configuration switch 306 to provide for the recovery of the management key as described in greater detail above.

[0059] Thus, methods and systems to support management operations associated with an interconnect device have been described. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these

embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.